## REMARKS

### Status of the Claims

- Claims 1-8 and 10-15 are pending in the Application after entry of this amendment.
- Claims 1-8 and 10-15 stand rejected.
- Claims 1 and 10 are currently amended by Applicant.

### Claim Rejections Pursuant to 35 U.S.C. §103

Claims 1-2, 5-7, 10-12, and 14-15 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Pat. Ser. No. 6,882,995 to Nasr in view of U.S. Pat. Ser. No. 6,785,673 to Fernandez et al.. Applicant respectfully traverses the rejection.

Nasr discloses a computer-implemented method of retrieving information in a first markup language through a query engine and presenting the information in any required markup language (Abstract).

The Office Action dated 7/12/2006 states on page 2 that Nasr teaches the XML intermediate language representation at col. 3 lines 10-18. Nasr at col. 3 lines 10-18 states:

> "In an embodiment shown in FIG. 1B, compiler 20 may receive a number of similar search requests in a number of differing languages, 5a, 5b, 5c, and 5d. The compiler 20 compile the search request 20 from any of the languages into the abstract engine language 25 and then have the abstract engine 30 run the search to obtain search result 15. The advantage is that the abstract engine can support any number of query languages."
> (col. 3 lines 10-18)

Applicant notes that this is the only reference in Nasr to the "abstract query language" itself. All other references appear to be directed to "Query Engine Instructions" as items 130 in Figure 1, item 220 in Figure 2, and item 330 in Figure 3. However, Applicant notes that very little description can be found of the features of the "Abstract Query Language or the Query Engine Instructions. Nasr states in col. 3 lines 28-33:

> "The Query Engine Abstract Machine 140 takes as input the following: Query Engine Instructions 130 and a Document Parse Tree 150 representation of a document. The query engine instructions tell the query engine what parts of the document parse tree to select and return as Query Results 160."
> (col. 3 lines 28-33).

Applicant notes that the query engine instructions (assumed to be the abstract engine language 25) are used along with a document parse tree 150 when used in the query engine abstract machine 140. Applicant concludes that the query engine instructions do not include a parse tree. Applicant also notes that the parse tree is a tree and not a graph. And that that query engine instructions do not include an explicit meaning of the input query as recited in amended Claim 1.

Amended Claim 1 recites:

> In a system for construction of executable queries, a method of communicating with an application using an application programming interface (API), the API method comprising:
>
> the system receiving from the application, one or more calls to set one or more compile parameters and commands for converting a plurality of input queries to an XML intermediate language representation, wherein the XML intermediate language representation is a composite of the plurality of input queries, is an explicit representation of the meaning of the plurality of input queries, and has a graph structure; and
>
> the system receiving from the application, one or more calls to convert the XML intermediate language representation to at least one executable query, the at least one executable query enabling the system to query over a plurality of data sources having differing data models.

Applicant has amended Claim 1 to further delineate that the method of communication is an API as described in conjunction with Figure 3 of the originally filed specification and that the XML intermediate language representation is not only a composite of the input queries, but is also an explicit representation of the meaning of the input queries and is also a graph structure as described in paragraph 0038 of the originally filed specification. Applicant notes that the XML intermediate language representation is not a tree structure.

Applicant notes that Nasr fails to disclose an API for the specific functions recited in Claim 1. Moreover, Applicant finds that the description of the "abstract engine language" or the "query engine instructions" are void of the combined characteristics of a composite of the plurality of input queries, an explicit representation of the meaning of the plurality of input queries and a graph structure as recited in amended Claim 1.

Accordingly, Nasr fails to recite all of the elements of amended Claim 1. In another sense, since the Nasr query engine instruction lacks a tree structure, but then adds a tree (not a graph) structure after the engine instructions are generated, Nasr teaches away from the XML intermediate language representation as defined by amended Claim 1.

Fernandez teaches at col. 2 line 62 through col. 3 line 22:

> The present invention overcomes many of the shortcomings of the prior art. In addition, the present invention addresses the problem of automating the conversion of relational data into XML. According to the invention, a general, dynamic, and efficient tool for viewing and querying relational data in XML referred to as SilkRoute is provided. SilkRoute is general, because *it can express mappings of relational data into XML* that conform to arbitrary XML schemas, not just a canonical mapping of the relational schema. *The mappings may be referred to as views.* Applications can express the data they need as an XML-QL query over the view. SilkRoute is dynamic, because it can materialize the fragment of an XML view needed by an application, and Silkroute is efficient, because it can fully exploit the underlying RDBMS (Relational DataBase Management Systems) query engine whenever data items in an XML view need to be materialized.
>
> According to one aspect of the present invention, a general framework is provided for mapping relational databases to XML views, to be used in data exchange. *In another aspect of the invention, a new query language, RXL, for mapping relational sources to XML views, is provided. According to yet another aspect, the present invention provides a sound and complete query composition algorithm that, when given an RXL query and an XML-QL query, generates a new RXL query equivalent to their composition.* In a still further aspect of the present invention, a technique is provided in which most of the work of an RXL query can be shipped to the underlying database engine.
> (col. 2 line 62 through col. 3 line 22).

Applicant notes that Fernandez teaches a new query language, RXL, for mapping relational sources to XML views. Also, Fernandez teaches that when given an RXL query and an XML-QL query as inputs, the Fernadez device generates a new RXL query equivalent to their composition as an output. This is shown in Figure 2 where an RXL input and a XML-QL input are provided to the query composer, a rewriter generates an new RXL query.

Applicant notes that this output of the query composer is a new query having the same basic form as the RXL input. A new and dramatically different representation form is not generated in the query composer of Fernandez.

Fernandez also teaches:

> In another aspect of the present invention, an intermediate representation of XML view queries called a view tree has been created that is general enough to express the XML mappings in any of these systems. An illustrative algorithm of the invention takes a view tree as input, and therefore could be directly applied to the XML view definitions expressed by these tools.
> (col. 3, lines 47-54.)

Applicant notes that a view tree (not a graph) is present in an intermediate representation of Fernandez. Thus, Applicant submits that Fernandez fails to disclose an XML intermediate language representation that is a composite of the plurality of input queries, is an explicit representation of  the meaning of the plurality of input queries, and has a graph structure as recited in amended Claim 1. Further, since Fernandez includes a view tree that is a tree and not a graph, then Fernandez, like Nasr, teaches away from amended Claim 1.

Applicant has amended independent Claim 10 to include the aspects of an XML intermediate language representation being a composite compiled from the plurality of input queries, the XML intermediate language representation is an explicit representation of  the meaning of the plurality of input queries, and the XML intermediate language representation having a graph structure.

Since both Nasr and Fernandez fail to teach all of the elements of independent Claims 1 and 10 then Nasr and Fernandez, considered alone or in combination, cannot render obvious Claims 1-2, 5-7, 10-12, and 14-15 because all elements are not taught or suggested by the combination of references as discussed above.

Accordingly, Applicant submits Claims 1-2, 5-7, 10-12, and 14-15 patentably define over the cited art. Applicants respectfully request withdrawal of the 35 U.S.C. §103(a) rejection as these claims patentably define over the cited art.

Claims 3 and 4 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Pat. Ser. No. 6,882,995 to Nasr in view of U.S. Pat. Ser. No. 6,785,673 to Fernandez et al.. and in further view of U.S. Pat. Ser. No. 6,961,728 to Wynblatt et al.. Applicant respectfully traverses the rejection.

Wynblatt teaches:

> Methods and a system that provide a network of data sources in terms of a traditional database schema, convert traditional database queries into network messages, and route those messages to those data sources which have relevant data, in some instances through designated query nodes. In the present invention, those data sources accept the message, and then send reply messages to the originator of the query, or via designated join nodes in some embodiments. In some embodiments, the data sources may be able to perform local join operations. The system then collects these reply messages as appropriate at the designated join nodes, designated query nodes, and/or at the data sources, and the reply messages that meet the requirements of the query are sent back to the query originator. The query results provided to the query originator are presented as a traditional database result. (Abstract)

Applicant notes that Wynblatt, like Nasr and Fernandez, fails to teach an XML intermediate language representation that is a composite of the plurality of input queries, is an explicit representation of the meaning of the plurality of input queries, and has a graph structure as recited in amended Claim 1 upon which Claims 3 and 4 depend. Accordingly, the combination of Nasr, Fernandez and Wynblatt cannot render obvious Claims 3 and 4.

Claims 8 and 13 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Pat. Ser. No. 6,882,995 to Nasr in view of U.S. Pat. Ser. No. 6,785,673 to Fernandez et al.. and in further in view of U.S. Pat. Ser. No. 6,934,712 to Kiernan et al.. Applicant respectfully traverses the rejection.

Kiernan et al. teaches:

> A method for publishing relational data as XML by translating XML queries into queries against a relational database. Conversion of the relational database into an XML database is not required. Each relational table is mapped to a virtual XML document, and XML queries are issued over these virtual documents. An XML query is parsed and transformed into a language-neutral intermediate representation, which is a sequence of operations describing how the output document is derived from the underlying relational tables. The intermediate representation is then translated into an SQL query over the underlying relational tables. The intermediate representation is also used to generate a tagger graph, which the tagger runtime 'walks' to generate the

> tagged, structured XML output. Each of the nodes of the tagger graph are operators which perform processing on the results of the SQL query. The SQL query is executed, and the SQL query results are then provided to the tagger. The tagger runtime applies the operators of each node to the inputs at that node to produce the structured XML document as a query result, guided by the structure of the tagger graph. (Abstract).

Applicant notes that Kiernan, like Nasr and Fernandez, fails to teach an XML intermediate language representation that is a composite of the plurality of input queries, is an explicit representation of the meaning of the plurality of input queries, and has a graph structure as recited in amended Claims 1 and 10 upon which Claims 8 and 13 depend respectively. Accordingly, the combination of Nasr, Fernandez and Kiernan cannot render obvious Claims 8 and 13.

## Conclusion

In view of the above remarks and amendments, Applicant respectfully requests withdrawal of the 35 USC §103 (a) rejections because the combination of any of Nasr, Fernandez, Wynblatt, and Kiernan cannot render obvious the amended independent claims and their respective dependent claims. Accordingly, Applicant respectfully requests reconsideration of all pending claims.

Respectfully submitted,

Date: August 23, 2006

Jerome G. Schaefer
Registration No. 50,800

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439